REAL-TIME NON-PHOTOREALISTIC RENDERING OF 3D CITY MODELS

H. Buchholz, J. Döllner, M. Nienhaus, F. Kirsch

University of Potsdam, Hasso-Plattner-Institut, 14482 Potsdam, Germany
(henrik.buchholz, juergen.doellner, marc.nienhaus, florian.kirsch)@hpi.uni-potsdam.de

KEY WORDS: Architecture, Cartography, Modeling, Reconstruction, Visualization, Building, City, Virtual Reality

ABSTRACT:

This paper presents a real-time non-photorealistic rendering technique that provides expressive depictions of 3D city models, inspired by the tradition of artistic and cartographic depictions typically found in bird's-eye view and panoramic maps. We define a collection of city model components and a real-time multi-pass rendering algorithm that achieves comprehensible, abstract 3D city model depictions based on edge enhancement and stylization, color-based and shadow-based depth cues, and procedural facade texturing. Non-photorealistic rendering facilitates the implementation of effective visual interfaces to urban spatial information and associated thematic information going beyond the Virtual Reality paradigm and offering a huge potential for graphics design. Primary application areas include city and landscape planning, cartoon worlds in computer games, and tourist information systems.

1. INTRODUCTION

1.1 Motivation

Depictions of virtual 3D city models, which integrate and visualize 2D and 3D geodata using the paradigm of the virtual city, involve manifold technical and artistic challenges. In the past, applications and systems of 3D city models have been based on concepts and techniques of Virtual Reality, implemented by a variety of real-time photorealistic rendering techniques.

This paper presents a new approach for depicting 3D city models based on concepts and techniques of non-photorealistic (NPR) rendering, a recently expanding, innovative genre in computer graphics. The presented NPR technique enables applications and systems to display large-scale 3D city models using expressive, stylistic graphics in real-time. It allows us to simulate classical, illustrative depiction techniques as well as to develop new techniques with high cognitive and perceptional qualities.

1.2 Historic Depiction Techniques of 3D City Models

Historic depiction techniques of 3D city models include bird's-eye view and panoramic maps. In general these representations are not only useful tools for effectively encoding, storing, and communicating urban spatial information but also turn out to be works of art, serving as valuable visual guides and visual indices to spatial structures and thematic information related to a city and its surrounding area. The abstracted, stylized presentations emphasize components of 3D city models and thereby ease recognition, facilitate navigation, exploration, and analysis of urban spatial information.

The roots of historic 3D city depictions date back over more than a thousand years. In pre-medieval times, depictions primarily revealed symbolic-allegoric contents, partially composed of landmark buildings but without adhering to topological and geometrical properties and relationships. Instead, topology and geometry were deduced from philosophical, mythical, or religious concepts. In medieval times, topology and geometry of urban areas were increasingly transferred to depictions, starting with orthogonal-like projections and being advanced by perspective drawings starting in the 16th century.

Matthäus Merian (1593-1650) is among the most prominent "city modelers": He established the first systematic production of city depictions as commercial products – manufacturing more than 2,150 European city views in his "Topographia", a book series with more than 30 volumes (Fig. 1a). The city views used elements of photorealism, but mostly abstracted and idealized the urban reality. In particular, using distortions and integrating

a)

b)

Figure 1: Example of a historic 3D city depiction, "The City of New York" by Guerber, 1870 (a). A typical bird's eye view map of the city of Rhede, Germany (b)

1. International Workshop on "Next Generation 3D City Models". Bonn, June 2006. In print.

multi-perspective views in a single image characterize the depictions.

Since that time, continuous advances in geodesy have been improving accuracy and completeness of the underlying data (Förstner 1999), while the general principles of city model visualization remained. Bird's-eye view maps are common for tourist information and description of local areas (Fig. 1b). Physical 3D city models, typically made of wood, represent common tools in city planning. In the 20th century, these models appeared for major cities, frequently abstracting from the photorealistic appearance of buildings while favoring representation of plan status or city development processes. Bollmann (1986) gives an overview of the production of 3D city maps from a cartographic point of view.

2. REAL-TIME NON-PHOTOREALISTIC RENDERING

2.1 Programmable Graphics Hardware

During the last years, an important trend in hardware design has been to increase programmability of graphics accelerators. Today's graphics hardware feature a partially programmable rendering pipeline and offer efficient parallel computing power. Graphics programming interfaces (e.g., OpenGL), offer shading languages that allow developers to formulate vertex and fragment programs used in the rendering pipeline by a high-level programming language (e.g., OpenGL Shading Language). Generally, multiple rendering passes and a variety of object-specific and effect-specific shaders are used to synthesize a single image.

Programmable graphics hardware represents the prerequisite for real-time rendering algorithms both in the fields of photorealism and non-photorealism. In particular, real-time enabled rendering techniques are required by any kind of interactive application. A general introduction to real-time rendering can be found in Akenine-Möller and Haines (2002).

2.2 Non-Photorealism

With the advent of *non-photorealistic computer graphics* a repertoire of illustrative, expressive, and artistic graphics techniques becomes viable to developers and designers of 3D applications and systems. Most researchers agree that the term "non-photorealistic" is not satisfying because the notion of realism itself is not clearly defined nor its complement, the non-photorealism. Nevertheless, non-photorealism (NPR) has established itself as a key category and discipline in computer graphics starting around 1990.

Non-photorealistic computer graphics denotes the class of depictions that reflect true or imaginary scenes using stylistic elements such as shape, structure, color, light, shading, and shadowing that are different from those elements found in photographic images or those elements underlying the human perception of visual reality. As Durand (2002) points out, non-photorealistic computer graphics offers extensive control over expressivity, clarity, and aesthetic, thereby the resulting pictures "can be more effective at conveying information, more expressive or more beautiful". Strothotte and Schlechtweg (2002) as well as Gooch and Gooch (2001) give a broad introduction to concepts and algorithms of non-photorealistic computer graphics. In general, NPR enables developers to present and depict visual information in a more purpose-oriented and task-oriented way using principles of classical illustration techniques.

2.3 Non-Photorealistic Rendering Techniques

The wealth of techniques comprise stylized digital halftoning for simulating handcrafted depictions, such as stippling, hatching, or engraving conveying illumination, curvature, texture, and tone in an image (Praun et al. 2001). Furthermore, techniques exist for generating technical illustrations automatically (Gooch et al. 1998) or for reproducing pencil or pen-and-ink drawings.

It has long been understood that just a "few good lines" (Sousa and Prusinkiewicz 2003) often suffice to encourage viewers to complete a picture by imagining the details that are missing. Object-space as well as image-space edge-detection algorithms exist that allow for producing line drawings (Isenberg et al. 2003).

Object-space edge-detection algorithms determine visually important edges in the 3D scene space and later resolve their visibility. They provide an analytic representation for edges so that they can be visualized by additional 3D scene geometry. This extra geometry represents stroke-like shapes that align loosely to the original geometry. Northrup and Markosian (2000) as well as Isenberg et al. (2002) implement real-time capable algorithms for edge detection and edge stylization. We apply an object-space edge stylization technique that has been designed for urban environments (Döllner and Walther 2003).

Image-space silhouette algorithms extract edges by way of image processing operators that detect discontinuities in specific image buffers, called G-Buffers (Saito and Takahashi 1990). These buffers store geometric properties of 3D geometries that result from conventional z-buffer rendering. Finally, visually important edges are represented in an image by pixels. Image-space algorithms can be fully accelerated by today's graphics hardware. Mitchell et al. (2002) present a first technique that extracts silhouettes for enhancing images on a per-scene basis rendering silhouettes and outline regions of shadow and texture boundaries.

Nienhaus and Döllner (2003) implement a similar algorithm that allows for distinctive rendering of a 3D scene's models by enhancing their visually important edges. For this, they extract discontinuities in the depth buffer and normal buffer and store them as edge intensities in a texture, called edge map. The edge map can be reused subsequently to enhance the 3D scene on a per-object basis.

The presented NPR rendering technique for 3D city models consists of specialized edge rendering techniques (enhanced and stylized edges), specialized shading techniques (n-tone shading) and combined real-time rendering techniques (shadowing and procedural texturing).

3. CHARACTERISTICS OF NPR CITY MODELS

In our approach, we aim at 3D city model depictions that are complementary to photorealistic renderings known, e.g., from Virtual Reality applications. Our approach aims at:

- Concentrate on illustrative, expressive visualizations emphasizing on high perceptual and cognitive quality to effectively communicate contents, structure, and relationships of urban objects as well as related thematic information;

- Enable meaningful visualizations even for the case of scarce urban spatial information since high-quality and complete data is rarely available for large-scale urban areas;
- Enable fully automated generation of visualizations while offering great flexibility in the amount of control of graphics design;
- Achieve real-time rendering and thereby allow for interactive manipulation, exploration, analysis, and editing of 3D city models.

Applications of non-photorealistic 3D city models are primarily all kinds of visual interfaces to urban spatial information required, for instance, in architectural drawings and sketches, city development planning and city information systems, radio and energy network planning, visualization of demographic development data, interactive gaming environments and comic worlds, and atmospheric and edutainment environments for narratives.

4. NON-PHOTOREALISTIC RENDERING ALGORITHM

The NPR rendering technique (Fig. 2) defines four rendering phases:

- Phase 1 renders shadow volumes and shadow regions;
- Phase 2 renders enhanced edges and textured facades;
- Phase 3 renders stylized edges;
- Phase 4 renders remaining components of a 3D city model.

4.1 Shadowing

Shadows in 3D city models are important depth cues and facilitate the perception of spatial coherence through the image. In city planning, shadows also represent a critical property of designs that needs to be visualized. To calculate shadows, we use a real-time implementation (Everitt and Kilgard 2002) of the *shadow volume technique* (Crow 1977).

In a pre-processing step, the illustrative visualization technique computes shadow volume geometry for a given 3D building geometry. In Phase 1 of our rendering algorithm, this geometry is used together with the 3D scene geometry to generate the stencil value zero for lit areas and non-zero for shadowed areas. For later use in shaders, the shadow information in the stencil buffer is copied into an alpha texture. To apply the texture in subsequent rendering phases, we create a screen-aligned quad that fits completely into the viewport of the canvas so that the texture coordinates (s,t) of each fragment produced for the quad correspond to windows coordinates.

For calculating shadows, we consider a single light source at infinite distance that is located above the horizon. Only building geometry casts shadows, i.e., shadow polygons are only determined for building geometry. This allows for optimized construction of shadow polygons: They consist of 1) quads, constructed by extruding roof border edges that face away from the light and 2) triangles extruded from wall edges of which one adjacent wall faces to, and the other away from the light. The polygons are extruded away from the light source and end below the ground plane. This approach does not require shadow polygons to be closed in contrast to shadow volume algorithms that are not specifically designed for city models. Therefore, fill-rate required to render the shadow polygons is lower, i.e., performance of our approach is superior.

4.2 Shading of Building Geometry

Cartographic city maps and other hand drawings of cities are using a reduced color scheme for shading. In colored drawings the illustrator usually abstracts from the realistic colors of the urban objects and greatly reduces the number of colors used. In general only two or three colors and for each color only two or three tints are used. The choice of the colors is based on aesthetic and other reasons as well as the actual colors of the city.

In Phase 2, we apply *n-tone shading* for all parts of buildings. First, the angle of the polygon's normal and the light direction is used to determine the intensity. Then, the intensity indexes a color palette of n tones determining the appropriate tone for shading. For this, each building is associated with a color palette defined by thematic information.

We observed that three or four tone shading supplies better results for interactive applications than two tone shading because there is much less contrast when moving through a scene and the light direction is much harder to determine.

To support depth cueing, we apply a linear transformation in tri-stimulus color space as described by Weiskopf and Ertl (2002). The saturation of a color is changed according to the viewer distance: more distant objects are rendered in more de-saturated colors whereas the intensities of colors remain constant.

4.3 Facades of Buildings

To cope with large-scale 3D city models, we do not create individual textures for each building. Instead, we combine texture elements to facades using multi-texturing. In a preprocessing step we create multiple sets of texture coordinates for each building, which are used by a fragment shader to compose the facade texture for each building. The facade composition is performed in 7 steps:

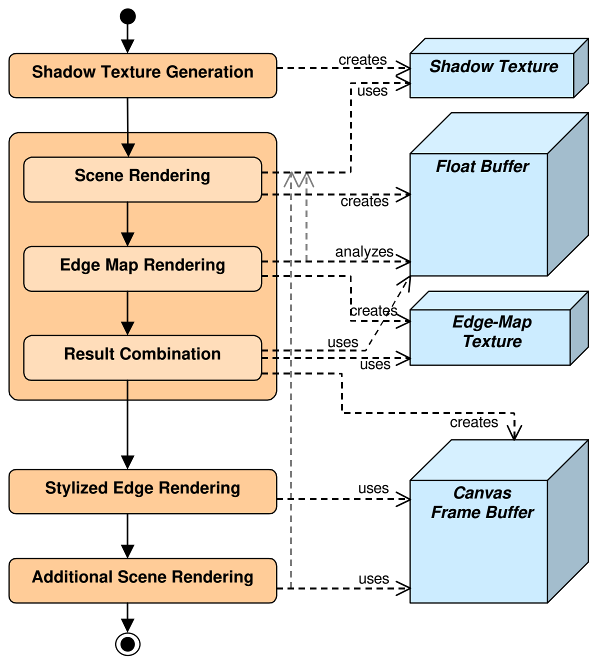


Figure 2: The non-photorealistic rendering pipeline for 3D city models, its principal rendering stages and graphics resources.

1. The material texture is wrapped around the facade with a uniform scale factor.
2. The window texture is projected as decal onto the textured facade. The texture coordinates are determined so that along a single wall, that is, from one corner of the building to the next, we get a whole number of adjoining windows. Decoupling window texture and material texture allows us to keep the material appearance undistorted.
3. The door texture is projected as decal onto the textured facade. For this, we create texture coordinates along the facade so that the *s* coordinate is in [0,1] where a door shall appear and outside [0,1] elsewhere. Using a door texture with transparent border, we obtain doors at the specified positions.
4. To prevent overlapping of doors and windows, we hide the windows near doors. Similar to step 3), we create additional texture coordinates, for which the *s* coordinate is in [0,1] where the windows shall be hidden.
5. Texturing the whole facade with a material texture leads to an unnaturally regular appearance and adds too much visual complexity to the resulting image. Inspired by hand-made drawings we blend between the facade color and the material texture so that the material is visible only at some irregular formed areas. These areas are defined by an additional alpha texture. In Fig. 3a), the areas can be clearly seen.
6. Texturing the whole facade with exact replicates of a single window texture leads to unnaturally regular appearance, too. We can avoid this by choosing each singular window randomly from a small set of variations of window textures. For example, we choose from well-lit and low-lit windows. For this, we use an alpha texture filled with random values. The texture coordinates for this texture are chosen so that each pixel is mapped on a single window of a building. The fragment shader uses the value of the alpha texture to choose from different window texture variants. The area of mapped pixels of the alpha texture is chosen randomly for each wall so that the arrangement of the alterations is different for each building. Fig. 3b) shows an example.
7. To reduce the visual complexity of the resulting image, the shader uses different levels of detail of window texture and door texture dependent on the viewer's distance. In contrast to mip-mapping, the use of explicit LOD textures allows us to consciously leave out details instead of compressing too much details into an unrecognizable size by filtering.

4.4 Image-Based Edge Enhancement

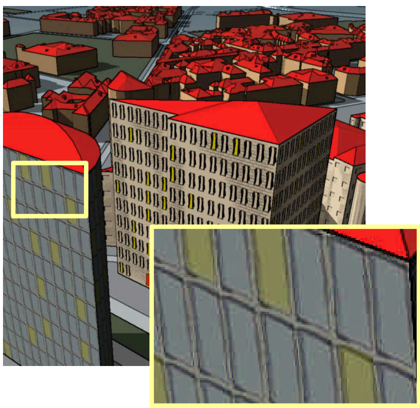
Image-space edge enhancement, part of Phase 2, detects visually important edges such as silhouettes, borders, and crease edges. The enhanced edges result in a homogenous and generalized visual depiction of 3D building geometry while emphasizing their principle composition.

Since the edge map algorithm is virtually independent from the number of polygons it is capable of handling 3D scenes having huge geometric complexity. Thus, image-space edge enhancement is particularly suitable for large-scale 3D city models but also applies well to the ground geometry, which generally contains a high number of curved shapes leading to a large number of short edges.

A typical characteristic of 3D city models is that a lot of faces are parallel to one another, such as buildings, roofs, and basement geometry. Furthermore, the scene measured from the front-most buildings to the buildings far away near the horizon is typically large with respect to depth, especially if viewed in a birds-eye-view. As a result, the normal buffer contains identical normal values for different faces and the depth buffer just shows a smooth depth gradient producing only minor changes in depth. Therefore, both buffers are not sufficient to adequately produce profile edges. For this reason, we must generate the ID buffer that encodes each single building as well as basement geometries by just an individual (*object identifier*) color value. In this way, we can extract profiles allowing for displaying both a building-to-building and a building-to-ground distinctively and for accentuating the outlines of roads, sidewalks, lawns, etc.

Thus, we obtain visually important edges by extracting discontinuities in the normal buffer, the ID buffer, and the z-buffer (Fig. 4). First, encoded normal, object identifier, and z-values of 3D scene geometry are rendered into a float buffer. Next, we extract discontinuities by rendering a screen-aligned quad, textured with the float-buffer texture and sampling neighboring texels to identify abrupt changes. The resulting image contains intensity values that represent edges of 3D scene geometry. The image is stored as texture, the edge map. Edge maps allow for enhancing edges with varying visual weight since they can be blended into 3D scene geometry.

In our examples, we reduce the "edge-ness" in the image with increasing camera distance. The edge-map technique is able to handle 3D scene geometry of any complexity; it is well suited for large-scale 3D data and virtually independent from the number of polygons. So, it is especially well suited for the ground geometry, which generally contains a high number of

a) 

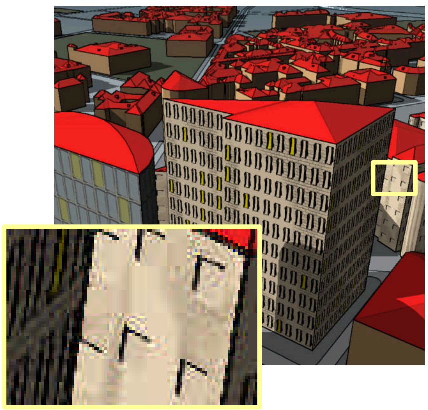
b) 

Figure 3: Material texture is shown at irregularly formed areas near edges (a). Different window variants (b).

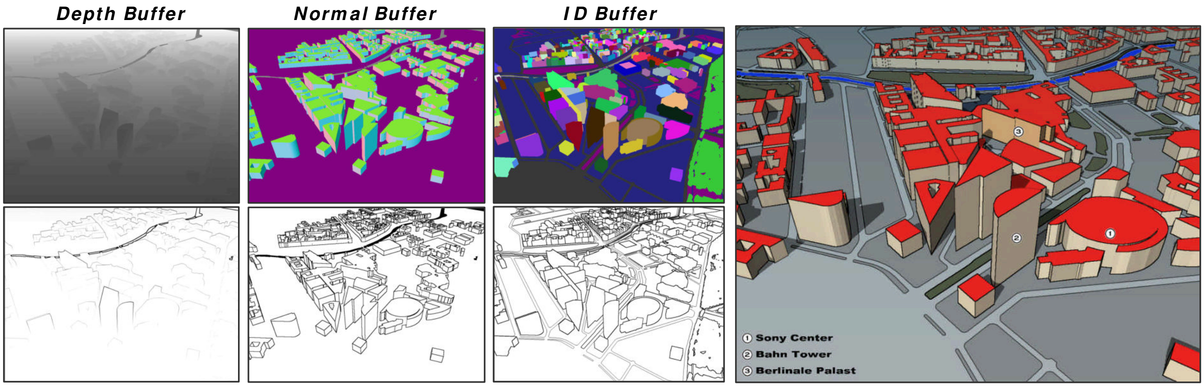


Figure 4: Discontinuities in the depth buffer, normal buffer, and ID buffer are the basis for edge-enhanced depictions of 3D city models. The examples show the Potsdamer Platz area, a part of the 3D city model of Berlin.

curved shapes leading to a large number of short edges. Furthermore, it does not require any pre-calculated information about topology and geometry of 3D objects.

4.5 Stylized Edges

For object-space edge stylization we use the algorithm described by Döllner and Walther (2003). It generates explicit 3D geomtry for visually important edges of buildings, which is textured similar to artistic edge strokes. The shape and appearance of stylized edges can be individually defined. In particular, stylized edges serve as tools to assign a characteristic appearance to visualizations and to depict specific thematic information such as planning state or renovation state.

To apply edge stylization to a building, a set of visually important edges must be detected first. They correspond primarily to hard edges of the generated geometry. However, some buildings have many small cavities, which lead to many hard edges that are not characteristic for the building's shape. So, we remove such cavities from the ground polygons in a preprocessing step: For each edge E_i we search in the succeeding edges for a collinear edge E_{i+k}, for which all vertices in between, that is the vertices of $E_{i+1}\ldots E_{k-1}$, are within an error-threshold distance from the line supported by E_i. If such an edge exists, we connect the first vertex of E_i with the second vertex of E_{i+k}, removing all intermediate vertices from the polygon.

The image-space edge technique enhances visually important edges in contrast to the object-space edge technique, which provides a high degree of freedom for graphics design. Together, they lead to an optimal tool for designing depictions of 3D building geometry.

4.6 Final Rendering Pass

Phase 4 renders all additional scene geometry such as 3D objects indicating scale, for instance, stylized humans or labels positioned in the view plane that provide texts or numbers referring to scene objects. In Fig. 5 an example of a real-time non-photorealistic depiction is given.

5. MODEL REQUIREMENTS

5.1 Building Models

The presented NPR rendering technique can process any kind of 3D building model, ranging from simple block models to detailed architectural models.

The only exception represents edge stylization: If a building model is given as an arbitrary geometric description, the automatic detection of important edges might not lead to useful results. In this case, the characteristic edges of a building must be indicated separately. We expect, however, that a heuristic approach can automated the detection in a future version.

5.2 Ground Space Models

The ground space includes mainly street space and green space, that is, those parts of the terrain surface not covered by ground plans. We distinguish and have experienced with two modeling approaches, image-based and geometric ground space representations.

- For image-based modeling of ground space, we can superimpose a 2D texture onto the terrain surface that encodes ground-space objects in the digital image, for instance, taken from aerial images.
- For geometric modeling of ground space, we can use 3D geometry, which represents the objects of the ground space such as sidewalks, lawns, etc. Extruding 2D polygons representing ground-space objects can generate this kind of geometry. The objects are laid upon the terrain surface.

For illustrative visualizations, we found that most common sources of data for image-based modeling, images derived from aerial photography, can hardly be transformed to an expressive or illustrative style – it contains too much photorealistic details. We prefer geometric modeling of ground-space objects because it can be derived from common 2D vector data of street systems and green-space areas. Geometric modeling seamlessly integrates into the non-photorealistic rendering process, that is, all rendering effects such as edge enhancement or depth cues are straightforward to apply. Insofar, ground-space objects are technically treated in the same way as buildings. Since the ground space is typically generalized to abstract from reality, the resulting objects can clearly be delineated.

6. CONCLUSIONS

Non-photorealistic rendering of 3D city models achieves expressive, stylized depictions, and it concentrates on 3D building and ground space geometry as most important elements of 3D city views. The technique can be applied even if relatively scarce input data is available, e.g., 2D ground plans, building heights, and additional associated thematic information.

We have demonstrated that successful visualization methods used in historical and traditional 3D city views can be transferred to the realm of interactive 3D computer graphics. Convincing illustrative 3D views require manifold illustrative design elements – the presented NPR technique provides a basis for implementing next generation 3D city model systems and applications.

The NPR rendering technique has been implemented as a part of the LandXplorer system, an authoring and presentation tool for 3D city models and 3D landscape models.

ACKNOWLEDGMENTS

We would like to thank our colleagues of the LandXplorer system (www.3dgeo.de), which has been used as implementation platform. We also would like to thank Falko Liecke, Senate of Economy Berlin, and Takis Sgouros, Senate of Urban Development Berlin, for their collaboration within the Berlin 3D City Model project, and A.P. Mulleners from Mulleners+Mulleners Architects and Urban Planner for giving us permission for Fig. 1b. We would like to thank the German environmental foundation Deutsche Bundesstiftung Umwelt for supporting our work within the Lenné-3D research project (www.lenne3d.de).

REFERENCES

Akenine-Möller, T., Haines, E., 2002. *Real-Time Rendering*. A K Peters Ltd, 2nd Ed.

Bollmann, F., 1986. Entstehung von Bildstadtplänen. *Karlsruher Geowissenschaftliche Schriften,* Reihe A, Band 4, 93-98.

Crow, F.C., 1977. Shadow Algorithms for Computer Graphics. *Computer Graphics (Proceedings of ACM SIGGRAPH '77)*, 11(2):242-248.

Döllner, J., Walther, M., 2003. Real-Time Expressive Rendering of City Models. Seventh International Conference on Information Visualization, Proceedings IEEE 2003 Information Visualization, 245-250, London.

Durand, F., 2002. An Invitation to Discuss Computer Depiction. *Symposium on Non-Photorealistic Animation and Rendering (NPAR).*

Everitt, C., Kilgard, M.J., 2002. Practical and robust stenciled shadow volumes for hardware-accelerated rendering. *NVidia Online Published Paper* (http://www.nvidia.com).

Förstner, W., 1999. 3D City Models: Automatic and Semiautomatic Acquisition Methods. *Proceedings Photogrammetric Week, University of Stuttgart,* 291-303.

Gooch, A., Gooch, B., Shirley, P., Cohen, E., 1998. A Non-Photorealistic Lighting Model for Automatic Technical Illustration. *Computer Graphics (Proceedings of SIGGRAPH 1998)*, 447-452.

Gooch, A., Gooch, B., 2001. *Non-Photorealistic Rendering.* A K Peters Ltd.

Isenberg, T., Freudenberg, B., Halper, N., Schlechtweg, S., Strothotte, T., 2003. A Developer's Guide to Silhouette Algorithms for Polygonal Models. *IEEE Computer Graphics and Applications,* 23(4):28-37.

Isenberg, T., Halper, N., and Strothotte, T., 2002. Stylizing Silhouettes at Interactive Rates: From Silhouette Edges to Silhouette Strokes, *Computer Graphics Forum (Proceedings of Eurographics)*, 21(3):249-258.

Mitchell, J., Brennan, C., Card, D., 2002. Real-Time Image-Space Outlining for Non-Photorealistic Rendering. *ACM SIGGRAPH 2002 Sketch.*

Nienhaus, M., Döllner, J., 2003. Edge-Enhancement - An Algorithm for Real-time Non-Photorealistic Rendering. *Journal of WSCG*, 11(2):346-353.

J. D. Northrup, J.D., Markosian, L., 2000. Artistic Silhouettes: A Hybrid Approach. *Proceedings of NPAR 2000.*

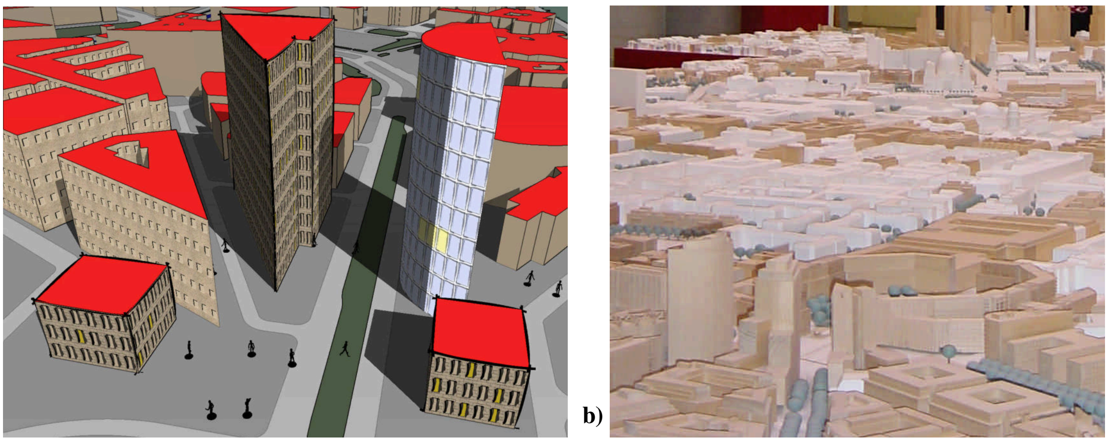
Praun, E., Hoppe, H., Webb, M., Finkelstein, A., 2001. Real-Time Hatching. *Computer Graphics (Proceedings of ACM SIGGRAPH 2001)*, 581-586.

Saito, T., Takahashi, T., 1990. Comprehensible Rendering of 3-D Shapes. *Computer Graphics (Proceedings of ACM SIGGRAPH '90)*, 197–206.

Sousa, M.C., Prusinkiewicz, P., 2003. A few good lines: Suggestive drawing of 3D models. *Proceedings of Eurographics 2003: Computer Graphics Forum*, 22(3):381-390.

Strothotte, T., Schlechtweg, S., 2002. Non-Photorealistic Computer Graphics: Modeling, Rendering and Animation. Morgan Kaufman.

Weiskopf, D., Ertl, T., 2002. Real-Time Depth-Cueing Beyond Fogging. *Journal of Graphics Tools*, 7(4):83-90.



a) **b)**

Figure 5: Example of a non-photorealistic depiction of the 3D city model of Berlin, a joint project with the Senate of Urban Development (left). Photo of the physical 3D city model of Berlin (right).